

## Claims

1. A method for data communication between processing machines, comprising:  
receiving a message produced by a first processing machine at a database server, the message being transmitted via socket protocol;  
analyzing the received message in the database server;  
storing the received message when the received message is processing data; and  
transforming the received message into a database instruction when the received message is an instruction.
2. The method of claim 1, wherein the message produced by the first processing machine includes a header formatted to identify whether the message is a database instruction.
3. The method of claim 2, wherein analyzing the received message in the database server includes analyzing the header of the message to determine if the message includes the database instruction.
4. The method of claim 1, wherein transforming the received message into a database instruction when the received message is the database instruction includes dynamically identifying an SQL instruction.
5. The method of claim 1, further comprising executing the instruction to produce a result.

6. The method of claim 5, further comprising:  
sending the result to the first processing machine.
7. The method of claim 6, wherein the message was produced as a result of processing within the first testing machine and the processing required the result to continue, and wherein the first processing machine receives the result set and resumes processing.
8. The method of claim 1, wherein the database server includes processing by a daemon.
9. The method of claim 8, wherein the daemon monitors a predetermined socket for messages from the first processing machine.
10. The method of claim 1, wherein the message produced by the first processing machine is transmitted to the database server as data included in the message that is produced.
11. The method of claim 1, wherein the message being transmitted via socket protocol includes being transmitted via TCP/IP protocol.
12. A system for communicating data between processing machines comprising:  
a database server;

at least one processing machine capable of communicating with the database server via a socket protocol;

a network for coupling the at least one processing machine to the database server; and

a daemon included within the database server, the daemon capable of determining if a message received from the at least one processing machine is a database instruction.

13. The system of claim 12, wherein the daemon is capable of monitoring a predetermined socket for the message from a first processing machine of the at least one processing machines.

14. The system of claim 12, wherein each of the at least one processing machine includes a corresponding operating system.

15. The system of claim 14, wherein the corresponding operating systems can be different for different processing machines, wherein the at least one processing machine is one of the different processing machines.

16. The system of claim 12, wherein a header of the received message includes an identification of the message as including the database instruction.

17. The system of claim 16, wherein the daemon executes dynamic language compiling.

18. The system of claim 17, wherein the dynamic language compiling includes SQL language compiling that has the capability of converting the message to an SQL instruction.

19. The system of claim 12, wherein the at least one processing machine is capable of communicating with the database server via a TCP/IP protocol.

20. A method for data communication between processing machines, comprising:  
receiving a message produced by a first processing machine at a database server, the message being transmitted via TCP/IP protocol and socket protocol, a header of the message being formatted to identify message as a database instruction, if the message is a database instruction;

analyzing the header of the received message through dynamic language compiling at the database server;

storing the received message when the received message is processing data;

transforming the received message into an SQL instruction when the received message is an instruction;

executing the instruction to produce a result; and

sending the result to the first processing machine.

21. The method of claim 20, wherein the dynamic language compiling monitors a predetermined socket for messages from the first processing machine.

22. The method of claim 20, wherein the message is transmitted to the database server as the data included in the message is produced in the first processing machine.